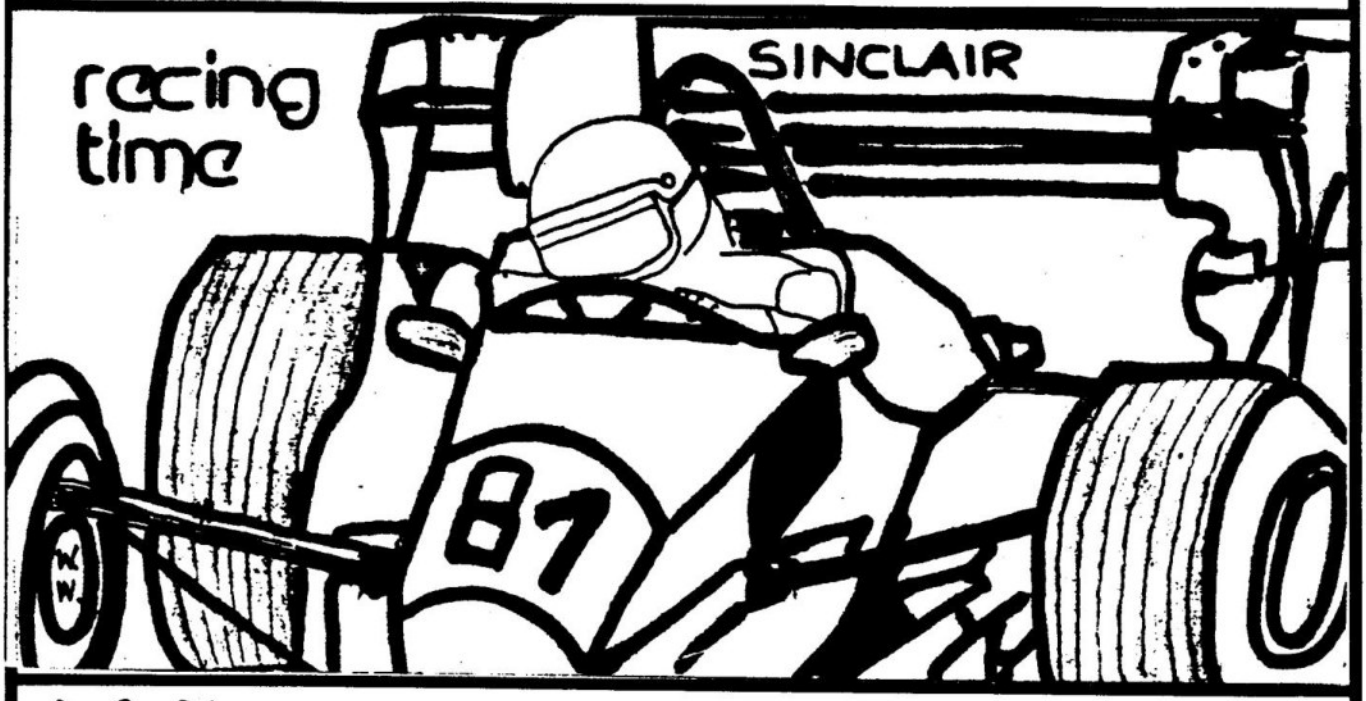
Spectrum User Club Wuppertal



Info Nr. 2 (74) 01. Februar 1990

Rolf Knorre, Postfach 20 01 02, 5600 Wuppertal 2

Hallo Freunde,

wie bereits im letzten Info angekündigt, habe ich die Clubkartei überarbeitet. Nach Löschung aller Mitglieder, die unter dem Weihnachtsbaum einen neuen Rechner gefunden haben, besteht der Club jetzt noch aus rund 130 Spectrum - Usern. Das ist seit Bestehen des Clubs ein absoluter Tiefpunkt. Leider wird sich daran wohl kaum noch etwas ändern lassen. Auf einige Anzeigen in der Happy Computer bzw. der ASM hat es so gut wie keine Resonanz gegeben. Trotzdem will ich den Club weiterführen, bis auch der harte Kern der jetzt noch vorhandenen Spectrum - Gemeinde aufgibt. Bei dieser Gelegenheit möchte ich den Mitgliedern danken, die Monat für Monat mit guten Artikeln dazu beitragen, daß das Info vom Umfang und Inhalt noch etwas hergibt.

Abschließend noch eine Anfrage: Wer steckt hinter der Bankverbindung Berliner Bank Konto Nr. 9944056300. Auf der Überweisung war kein Absender angegeben.

Viel Spaß mit dem neuen Info wünscht

Rolf Knorre

Spectrum User Club Wuppertal

Wilhelm Wiecke, Lönsweg 92, 4904 Enger Tel. 05224/3449

Folge 9

Enger, den 7.1.90

Hallo Spectrum-User!

Nun will ich versuchen, meinen ersten Beitrag im neuen Jahr für's Club-Info zu schreiben. Hoffentlich bleibt die Mehrzahl der Mitglieder auch 1990 dem SPECTRUM treu und damit hoffentlich auch dem Club! Ich werde jedenfalls noch lange den SPECTRUM nutzen, denn bisher habe ich seine Leistungsfähigkeit noch lange nicht voll ausgeschöpft! Da mich auch besonders das Programmieren und Analysieren von Maschinensprache-Programmen interessiert, bieten sich noch reichlich Möglichkeiten. Schließlich kann der SPECTRUM mit intelligent gemachten Programmen und Peripherie fast alles, wenn auch langsamer und vielleicht nicht immer so komfortabel, was wesentlich teurere Rechner leisten! Daher kann ich dem Beitrag von Herrn Hartig im Info 11/89 und Herrn Abel (Info 12/89) nur zustimmen, denn ich will schließlich nicht nur 'Anwender' sein, sondern auch noch verstehen, was da abläuft.

Jetzt soll's aber endlich weitergehen mit meiner Serie zur BETA-Disk-Version 5.03! Letztes Mal habe ich versucht, die wichtigsten Funktionen (zur Erinnerung: sie dienen dem Aufruf verschiedener Befehle aus MC-Programmen) zu erläutern. In dem erläuterndem Text zur Funktion 14 (Bytes laden) ist mir jedoch ein kleiner Fehler unterlaufen, weil ich das folgende Programm später nochmals umnummeriert habe. Daher muß es im Text statt Zeile 75 nun 95 bzw. statt 220 jetzt 240 heißen. Wer genauer hingesehen hat, wird's sicherlich schon gemerkt haben!

Inzwischen ist also geklärt, was die Funktionen 1, 5, 6, 7, 11, 12, 14 und 18 bewirken und wie sie sich ansprechen lassen. Wie sieht's aber bei den Übrigen aus? Also, Funktion 13 und 15 - 17 können wir sofort abhaken, den sie bewirken nur eine Rückkehr ins BASIC. Hier kann man bei einem geänderten EPROM eventuell eigene Aufrufadressen einbauen und damit eigene Funktionen aufrufen.

Von großer Bedeutung ist Funktion 24! Sie bewirkt unter anderm, daß gegebenfalls der Puffer erweitert und der Organisationssektor geladen wird. Danach findet eine Überprüfung des Disk-Formates mit Anpassung der entsprechenden Systemvariablen. Diese Funktion muß immer vorgeschaltet werden, wenn sichergestellt werden muß, das unterschiedliche Disk-Formate erkannt werden. Das ist eigentlich bei allen Schreib- und Lese-Operationen auf und von Disk erforderlich. Bei einigen Funktionen wird dies schon automatisch berücksichtigt (z.B. bei den Funktionen 7, 11, 12 und 18). Bei Nummer 14 (Bytes laden) ist dies jedoch nicht der Fall, deshalb wurde es im entsprechenden Programm (Info 12/89) in Zeile 130-140 nachgeholt! Näheres zu Funktion 24 ist dem Assemblerlisting in Folge 6 dieser Serie zu entnehmen (Einstiegsadresse $0405)!

Funktion 0 dient hauptsächlich der BREAK-Abfrage. Da sie fester Bestandteil der Schreib- und Lese-Operationen ist, kann man gewöhnlich auf einen gesonderten Aufruf verzichten.

Die auch noch nicht behandelte Funktion 2 bewirkt eine Einstellung des angesprochenen Laufwerks auf eine im A-Register übergebene Tracknummer! Auch diese wird praktisch bei allen Schreib-/Lese-Zugriffen genutzt. Eigene Verwendung habe ich bisher noch nicht dafür gehabt. Wer will, kann ja mal folgende kleine Routine per Assembler eingeben und testen:

```
CALL $3D21    ;TRDOS-Puffer testen bzw. installieren
LD A,33       ;Tracknummer ins A-Reg. schreiben
LD C,2        ;Funktionsnummer übergeben
CALL $3D13    ;Funktion aufrufen
RET           ;Zurück ins BASIC
```

Bei Aufruf wird nun der Schreib-/Lese-Kopf auf Track 33 eingestellt! Andere Tracknummern können natürlich im Rahmen des technisch Möglichen eingegeben werden (also bei 40 Track-Laufwerk 0-39, bei 80 Tracks 0-79 für einseitige und 0-79 bzw. 0-159 bei doppelseitigen).

Die Funktionen 3 und 4 sind meiner Meinung nach überflüssig, da ihr Aufruf platzsparender durch direkte Ladebefehle ersetzt

werden kann! Bei Nummer 3 wird nämlich nur eine Sektornummer aus dem A-Reg. in die Systemvariable $5CFF übergeben, während Funktion 4 eine Adresse aus dem HL-Register nach $5D00 schreibt!

Funktion 10 vergleicht einen im TRDOS-Systemvariablenbereich ab $5CDD abgelegten Programmnamen der Reihe nach mit denen in der zuvor im erweiterten Puffer geladenen Katalog-Kopie. Wenn er gefunden wird, steht im C-Reg. die Nummer des gefunden Namens im Katalog. Wird diese nun ins A-Reg. übergeben, kann durch Aufruf von Funktion 2 oder 9 daraus der richtige Sektor errechnet und dann der komplette Datensatz des entsprechenden Programms bei 8 aus der CAT-Kopie in den TRDOS-Puffer und bei 9 in umgekehrte Richtung kopiert werden. Praktisch angewendet habe ich dies bisher noch nicht in eigenen Routinen, daher erfolgen diese Angaben nur unter Vorbehalt!

Letzteres gilt auch für Funktion 19 und 20! Soweit zu übersehen, läßt sich auch damit der zu jedem Programm im Katalog gehörende Datensatz (bestehend aus Namen, Typ, Startadresse, Blocklänge, Fileanzahl, Startsektor und Starttrack, vergleiche dazu Info 9/89) mit Funktion 19 von einer im HL-Reg. übergebenen Adresse in den TRDOS-Systemvariablenbereich nach $5CDD-$5CEC kopieren. Funktion 20 erlaubt den Transfer der Programmparameter in die umgekehrte Richtung. Damit kann man eventuell eine Routine stricken, die es erlaubt, den Kataloginhalt in jeden gewünschten Speicherbereich zur weiteren Bearbeitung zu laden. Das läßt sich vielleicht nutzen, um die Parameter z.B. für die Verwendung in einem Textverarbeitungsprogramm aufzubereiten, wo sich dann auch weiterführende Kommentare einbauen ließen für detailierte Programmlisten usw.! Der Fantasie sind also keine Grenzen gesetzt, nur ist die Realisierung der Ideen meist sehr zeitraubend!

Bleiben noch die Funktionen 21 - 23! Nummer 21 dient zur Überprüfung der einzelnen Tracks auf Fehlerfreiheit, Funktion 22 wählt die Seite 0 einer Disk an und 23 entsprechend Seite 1. Das ist ja erforderlich, sobald bei einer doppelseitigen Diskette eine Tracknummer >39 (bei 40 Trk DS) oder >79 (bei 80 Trk DS) gelesen oder beschrieben werden soll. Gewöhnlich geschieht das Umschalten zwischen beiden Seiten unbemerkt, da automatisch.

Wie die letzten 3 Funktionen und einige schon genauer beschriebene genutzt werden können, geht aus dem kommentierten Assemblerlisting des Programms "doctor" von der Utility-Disk der BETA 5.03-Version hervor. Dies bietet ein anschauliches Beispiel, wie sich mehrere Funktionen übersichtlich und platzsparend aufrufen lassen!

Wer allerdings den BASIC-Teil von 'doctor' und den MC-Teil vergleicht, wird sofort auf Unstimmigkeiten stoßen. In den REM- Zeilen 40 bis 60 im BASIC-Programm werden nämlich falsche Adressen angegeben! Korrekt muß es dort heißen:

```
40 REM 60015 sector      (statt 60014)
50 REM 60016 track       (statt 60015)
60 REM 60017 drive no    (statt 60016)
```

Praktische Auswirkungen hat der Fehler aber nicht, da im Programm selbst die richtigen Adressen benutzt werden. Aber es kann irritierend wirken! Übrigens möchte ich nochmals darauf hinweisen, daß es ratsam ist, zuerst nach Einlegen einer zu untersuchenden Diskette Menu-Punkt 8 (Change disk) aufzurufen. Meine Utility-Disk ist nämlich mit 80 Trk SS formatiert und darauf werden auch die TRDOS-Systemvariablen eingestellt. Wenn die neu eingelegte Disk jedoch in einem anderen Format formatiert ist, werden die übrigen Menu-Punkte nicht korrekt behandelt. Das richtige Format wird erst durch 'Change disk', was exakt einen Aufruf der Funktion 24 bedeutet, eingestellt!

Wie im Listing zu sehen, werden alle Funktionen über JUMP-Befehle (JP xxxx) erreicht, die wiederum aus dem BASIC jeweils mit 'RANDOMIZE USR xxxxx' angesprungen werden. Weil alle benutzten Routinen und Einsprünge in der sogenannten Sprungleiste stehen, ist das Programm sehr 'pflegeleicht', das heißt, leicht zu erweitern und zu ändern. Für eigene MC-Programme sollte man sich diese Vorgehensweise merken, denn nebenbei wird's auch noch übersichtlicher!

Nun aber erst mal das Assemblerlisting:

```
EA60            0000          ORG  60000       ;Startadresse von CODE 'doctor'
                0010 ;
EA60 C38DEA     0020          JP   readsek     ;Aufrufadr. 60000 für Funktion 5 (= Sektor lesen)
EA63 C393EA     0030          JP   writsek     ;Aufrufadr. 60003 für Funktion 6 (= Sektor schreiben)
EA66 C399EA     0040          JP   seldriv     ;Aufrufadr. 60006 für Funktion 1 (= Laufwerk wählen)
EA69 C3A2EA     0050          JP   chadisk     ;Aufrufadr. 60009 für Funktion 24 (= Change Disk)
EA6C C3A8EA     0060          JP   veritrk     ;Aufrufadr. 60012 für Funktion 21 (= Verify komplette Disk)
                0065 ;
EA6F            0070 seknr    DB   $00         ;Platz für Sektornummer (Wert 0-15)
EA70            0080 trknr    DB   $00         ;Platz für Tracknr. (Wert 0-39/79/159 je nach Diskformat)
EA71            0090 drivenr  DB   $00         ;Platz für Drivenummer (Wert 0-3)
                0095 ;
EA72 C3AEEA     0100          JP   side0       ;Aufrufadr. 60018 für Funktion 22 (= Disk-Seite 0 wählen)
EA75 C3B4EA     0110          JP   side1       ;Aufrufadr. 60021 für Funktion 23 (= Disk-Seite 1 wählen)
                0115 ;
EA78 CD133D     0120 runfunk  CALL $3D13       ;Gewählte Funktion ausführen!
EA7B C9         0130          RET              ;Zurück ins BASIC!
                0140 ;
EA7C 0601       0150 sekload  LD   B,1          ;Zu ladende bzw. speichernde Sektorenzahl ins B-Reg. und
EA7E 219CEB     0160          LD   HL,$EB8C    ;Zieladr. bzw Startadr. (hier 60300) dafür in HL laden!
EA81 3A70EA     0170          LD   A,(trknr)   ;Eingegebene Tracknummer ins A-Reg. einlesen
EA84 57         0180          LD   D,A         ;und ins D-Reg. übertragen!
EA85 3A6FEA     0190          LD   A,(seknr)   ;Eingegebene Sektornummer ins A-Reg. einlesen
EA88 5F         0200          LD   E,A         ;und ins E-Reg. bringen!
EA89 CD133D     0210          CALL $3D13       ;Gewählte Funktion ausführen!
EA8C C9         0220          RET              ;Zurück ins BASIC!
                0230 ;
EA8D 0E05       0240 readsek  LD   C,5         ;Funktionsnr. 5 laden (= Sektor lesen)
EA8F CD7CEA     0250          CALL sekload     ;Parameter laden und Funktion ausführen.
EA92 C9         0260          RET              ;Zurück ins BASIC!
                0270 ;
EA93 0E06       0280 writsek  LD   C,6         ;Funktionsnr. 6 laden (= Sektor auf Disk schreiben)!
EA95 CD7CEA     0290          CALL sekload     ;Parameter laden und Funktion ausführen.
EA98 C9         0300          RET              ;Zurück ins BASIC!
                0310 ;
EA99 3A71EA     0320 seldriv  LD   A,(drivenr) ;Eingegebene Laufwerksnummer ins A-Reg. holen
EA9C 0E01       0330          LD   C,1         ;und Funktionsnr. 1 laden (= Laufwerksnr. wählen)
EA9E CD78EA     0340          CALL runfunk     ;und 'Funktion ausführen' aufrufen!
EAA1 C9         0350          RET              ;Zurück ins BASIC!
                0360 ;
EAA2 0E18       0370 chadisk  LD   C,24        ;Funktionsnr. 24 laden (= Diskformat überprüfen)
EAA4 CD78EA     0380          CALL runfunk     ;und 'Funktion ausführen' aufrufen!
EAA7 C9         0390          RET              ;Zurück ins BASIC!
                0400 ;
EAA8 0E15       0410 veritrk  LD   C,21        ;Funktionsnr. 21 laden (= Verify track)
EAAA CD7CEA     0420          CALL sekload     ;Parameter laden und Funktion ausführen.
EAAD C9         0430          RET              ;Zurück ins BASIC!
                0440 ;
EAAE 0E16       0450 side0    LD   C,22        ;Funktionsnr. 22 laden (= Seite 0 der Disk wählen)
EAB0 CD78EA     0460          CALL runfunk     ;und 'Funktion ausführen' aufrufen!
EAB3 C9         0470          RET              ;Zurück ins BASIC!
                0480 ;
EAB4 0E17       0490 side1    LD   C,23        ;Funktionsnr. 23 laden (= Seite 1 der Disk wählen)
EAB6 CD78EA     0500          CALL runfunk     ;und 'Funktion ausführen' aufrufen!
EAB9 C9         0510          RET              ;Zurück ins BASIC!

#      61A8  seknr   EA6F  trknr   EA70  drivenr EA71  runfunk EA78  sekload EA7C  readsek EA8D  writsek EA93  seldriv EA99
chadisk EAA2  veritrk EAA8  side0   EAAE  side1   EAB4
```

Ich vermute, daß es auch für ältere BETA-Versionen einen Diskettenmonitor in ähnlicher Form gibt, der wahrscheinlich sogar nicht viel anders aufgebaut ist als dieser hier. Es lohnt sich bestimmt, einmal nachzusehen, bietet sich doch damit die Möglichkeit, etwas über die Funktionen und deren Aufrufadresse zu erfahren.

Wer übrigens mehrere Sektoren gleichzeitig laden will um sie sich z.B. mit Menu-Punkt 6 (Display sektor) anzusehen, kann dies leicht haben! Dazu braucht nur in Adresse $EA7D ein neuer Wert mit 'POKE 60029,x' eingegeben werden. Der Wert x sollte nicht größer als 16 sein, womit ein ganzer Track in einem Rutsch geladen (oder auf Disk zurückgeschrieben) wird! Eventuell kann man sich eine entsprechende Abfrage auch in den BASIC-Teil einbauen, etwa so:

```
9205 GOSUB 9400      bzw  9305 GOSUB 9400
......
9400 INPUT "Sektoranzahl? (1-16) ";sz : IF sz < 1 OR sz > 16
THEN GOTO 9400
9410 POKE 60029,sz :RETURN
```

Es bedarf einer weiteren Änderung und zwar in Zeile 8420, damit die richtige Sektorenanzahl auch gelistet werden kann:

```
8420 FOR i=a TO 255 * sz STEP 4
```

Dann läßt sich jedesmal die Anzahl neu festlegen. Das gleichzeitige Laden mehrerer Sektoren spart bei Sucherei auf der Disk viel Zeit! Vorsicht ist allerdings besonders beim Zurückschreiben von Sektoren auf Disk geboten! Dann sollte die eingegebene Sektorenzahl keinesfalls größer als die zuvor beim Lesen verwendete sein. Andernfalls werden eventuell wichtige Daten mit 'Müll' überschrieben und das kann unter Umständen die ganze Disk oder zumindest Teile davon unbrauchbar machen!

Natürlich gibt's noch viele Verbesserungmöglichkeiten, so könnte man sich, während der Sektorinhalt gelistet wird, ständig die zugehörige Track- und Sektornummer mit anzeigen lassen usw.! Das erspart einem dann hinterher eventuell die Rechnerei, herauszufinden, in welchem Track und Sektor man gerade liest. Dazu vielleicht ein anders Mal mehr!

Bei Menu-Punkt é ist Ausgabe auf einen Drucker manchmal wünschenswert! Das ist eigentlich kein großes Problem, da man im Grunde ja nur von Kanal 2 auf Kanal 3 umschalten muß. Folgende Änderungen sind dazu notwendig:

```
8402 PRINT "2) Bildschirm  3)Drucker? ";ch: IF ch<2 OR ch>3 THEN
GOTO 8402
```

Nun muß noch in Zeile 8040, 8100, 8120, 8336, 8360, 8365, 8405, 8430 und 8445 der PRINT-Befehl ergänzt werden zu 'PRINT #ch;' und Zeile 8460 wird folgendermaßen erweitert:

```
8460 LET ch=2: PRINT #ch; "press key for menu";: PAUSE 0: PAUSE
0: RETURN
```

Die Initialisierung des Druckers sollte etwa in einer Zeile 202 geschehen. Bei meinem Kempston E-Interface brauche ich dazu nur

```
202 COPY: REM /1
```

eingeben (= Line feed und Cariage return on)! Hier können auch noch andere Befehle an den Drucker, z.B. Zeilenabstand oder Schrift ändern, eingebaut werden. Natürlich läßt sich das auch erst ab Zeile 8400 einbauen, wenn man will. Die Beispiele sollen schließlich nur als Anregung und kleine Starthilfe dienen!

Da das Handbuch leider kaum Angaben zur Handhabung des Disk-Monitors macht, noch ein paar Tips zur Adresseneingabe:

Wenn bei Menu-Punkt ó bzw. 7 die Startadresse abgefragt wird, hat man mehrere Eingabemöglichkeiten und zwar dezimal oder hexadazimal. Letzteres wird durch ein nachgestelltes 'H' oder 'h' gekennzeichnet! Dazu Beispiele:

```
Dezimal     :  127
Hexadezimal:  7fh, 7FH, 7fH, 7Fh, 07fh, 007Fh usw.
```

Es können also Hex-Zahlen mit 1-4 Stellen angegeben werden in beliebiger Mischung von Groß- und Kleinschrift! Zu beachten ist lediglich, daß Startadresse nicht größer sein sollte als 'Sektorzahl * 256'. Beim Ändern von Sektorinhalten mit Menu-Punkt 7 werden nur Hex-Zahlen akzeptiert, zweistellig und ausnahmsweise ohne nachgestelltes 'h'!

Um die geänderte Version des 'doctor' abzuspeichern, kann man noch folgende Zeilen anhängen:

```
9990 RANDOMIZE USR 15619: REM: SAVE "doctor" LINE 5
9991 RANDOMIZE USR 15619: REM: SAVE "secread" CODE 60000,90
```

Mit RUN 9990 aufgerufen, wird die neue Version am Besten auf eine neue Disk gespeichert. Es empfiehlt sich nicht, die Originaldisk dafür zu benutzen, denn die sollte immer schreibgeschützt in Reserve bleiben.

Nun Schluß für diesen Monat! Wenn die Zeit reicht, habe ich vor, daß nächste Mal u.a. zu erläutern, wie man gelöschte Files wieder retten kann und anderes mehr. Bis dann!

Spectrum User Club Wuppertal

Frank-Michael MOCZKO

Haselsteig 41
1000 Berlin 47

Berlin,25.10.89

Liebe Spectrum - Freunde ,

im Roman "TIM Thaler" gibt es eine Hauptfigur Baron " LeFuet". Mit ein bißchen Phantasie kann man das Wort umdrehen und erhält "Teufel".

Mit einem einfachen HiSoft - PASCAL - Programm will ich nun erreichen, daß ein Wort , hier aus sechs Buchstaben , eingelesen wird und dann rückwärts wieder ausgegeben wird:

```
10    PROGRAM TIM;

20    Var Z1,Z2,Z3,Z4.Z5,Z6 : Char;
30    BEGIN

40    WRITELN(´ Geben Sie ein Wort mit sechs Buchstaben ein ´);
50    READLN(Z1,Z2,Z3,Z4,Z5,Z6);
60    WRITELN(Z6,Z5,Z4,Z3,Z2.Z1)

70    END.
```

Soweit , so gut : aber beim Ablauf kommt nur Unsinn heraus !!

Sehen wir uns einmal allgemein die READLN - Prozedur an :

Wenn die Variable Z1 eine INTEGER -Zahl ist , z.B.: 124 , Z2 z.B.: 567 , gibt es keine Probleme ; man tippt die Zahl Z1 ein , macht ein <SPACE> als Trennoperator ,dann Z2 , <SPACE> ,.... ,am Ende zweimal <ENTER> , um die Eingabe abzuschließen. (Das zweimalige ENTER entspricht nicht dem STANDARD - PASCAL).

Für REAL -Zahlen gilt das sinngemäß gleiche.

Anders aber bei CHAR - Variablen , wo ja jede Taste , also auch die <SPACE> -Taste , als ein Zeichen gilt.

Hier verwendet man folgendes Verfahren:

a) Vor der eigentlichen READLN - Zeile kommt eine Zeile, in der nur READLN; steht. (In unserem Beispiel : Zeile 45)
b) Beim Ablauf gibt man die Zeichen (z.B.: TEUFEL) dann ohne <SPACE> ein.
c) Abschließen der Eingabe durch zweimaliges <ENTER>.

Bis zur nächsten PASCAL - Sitzung dürft Ihr Euch mal überlegen , wie man die lästige Eingabemethode Z1,2...N mit einer Schleife eleganter lösen kann.

Aus Berlin mit vielen Grüßen

Michael

Spectrum User Club Wuppertal

Liebe Clubfreunde,

nachdem ich nun seit einem Jahr Mitglied im User-Club bin, möchte ich endlich auch einmal einen Beitrag leisten.

Da ich sehr viele Akkus für Kinderspielzeug und meine eigenen Spielzeuge (Fernlenkautos) zu laden habe, bin ich auf den Gedanken gekommen, die überwachung des Ladevorganges meinem Spectrum zu übertragen. Ich habe dazu eine Schaltung entwickelt, aufbauend auf einer Schaltung aus dem " ZX-Spectrum Hardware Handbuch ": 8-Kanal Analog-Digital Konverter. Bei mir hat sich die Schaltung schon im praktischen Einsatz bewährt.

Ein Sinterzellen-Akku kann problemlos sehr viele Schnelladungen vertragen, vorausgesetzt er wird nicht überladen. Und hier liegt das Problem. Wird ein Akku überladen, so wandelt er die zugeführte Energie nur noch in Wärme um. Dann passiert es, daß die Zelle an Kapazität verliert und der Akku unbrauchbar wird. Um ein Überladen zu verhindern, muß der Akku vor dem Laden auf seine Entladespannung gebracht werden. Anschließend wird der Akku 35 bis 40 Minuten mit 20-25 * I/10 geladen.
Dabei erreicht man Ladekapazitäten von 70% bis 80%. Sinnvoller ist es den Ladezustand "voll" festzustellen und genau in diesem Moment den Ladevorgang abzubrechen. Dazu muß man wissen, daß die Zellenspannung eines NC-Akkus beim Laden zuerst ansteigt, bei Erreichen des Ladezustands voll=100% aber wieder abfällt. Die Abschaltautomatik muß diesen Zeitpunkt eindeutig bestimmen. Dazu wird die Zellenspannung in regelmäßigen Abständen gemessen.

Grundlage dazu ist das A/D-Wandler-IC ZN427E. Der Anschluß der Schaltung erfolgt über die Z 80 A PIO. Als Taktgeber für den Wandler dient ein CD4011BE-IC. Um möglichst kleine Spannungsunterschiede messen zu können, kann die Anzahl der angeschlossenen Akkus mit einem Dreistufenschalter vorgewählt werden. Die Versorgungsspannungen bezieht die Schaltung vom Spectrum. Die -5V werden am Userport 20A abgegriffen, die +5V werden mittel eines Festspannungsreglers aus den +9V (4B) erzeugt, da die Spectrum +5V nicht genügend Strom liefern können. Der Spannungsregler muß auf ein kleines Kühlblech montiert werden.

Zum Basic-Listing ist nicht viel zu sagen, da es sehr viele Rem-Zeilen enthält. Die Parameter für den Ladevorgang sind so eingestellt, daß diese für die meißten Schnelladevorgänge geeignet sind. Es ist aber möglich die Parameter für Anfangsnormierzeit, Anzahl gleicher Messungen zum Abschalten und Abtastintervall zu verändern.

Es sollte zu Beginn einer Schnelladung eine gewisse Zeit ohne aktivierte Abschaltautomatik ablaufen. Es können größere Spannungsunterschiede durch Formierungsprozesse innerhalb der Zellen auftreten, welche zu Fehlabschaltungen führen.

Um eine gewisse Sicherheit gegen Fehlabschaltungen zu erreichen, sollten mehrere gleiche Messungen zur Abschaltung eingestellt sein. Wird die Anzahl gleicher Messungen erhöht, müssen die Abtastintervalle verkürzt werden.

Zum Einstellen der fertigen Schaltung dient das kurze Listing 2. Zuerst wird getestet ob das Relais richtig schaltet. Anschließend kann die Akkuspannung eingestellt werden. Dazu werden 4 volle Akkus angeschlossen. Zum Messen der Akkuspannung sollte ein Digitalvoltmeter verwendet werden. Die Spannung der vier vollen Akkus beträgt etwa 5,25 Volt. Am Port B des Pio Bausteins stehen je nach Schalterstellung ca-Werte von 170, 120 und 60. Mit dem jeweils zur Schalterstellung gehörenden Trimmer werden die Volt-Werte eingestellt, welche das Voltmeter anzeigt. Die eingestellten Volt-Werte sollten möglichst gleich sein. Wenn bis hier alles in Ordnung ist, dürfte die Schaltung funktionieren.

Spectrum User Club Wuppertal

Stückliste:

Halbleiter:

4 LED rot
1 Transistor BC 238 B o.ä.
1 Diode 1N4148

ICs: jeweils mit Sockel

1 Z80A PIO
1 CD 4011 BE
1 ZN 427 E (A/D Wandler)
1 7805 (Spannungsregler positiv)

Trimmpotis:

2 10 K
1 50 K

Sonstiges:

1 Userport-Stecker
1 Veroplatte (?)
1 Drehschalter, 4 Ebenen, 3 Stellungen
1 kleines Kühlblech
1 Siemens Kartenrelais 5 Volt 1*Um
Fädeldraht (?)

Kondensatoren:

1 22 myF Elko
1 100 myF Elko
1 1 myF
1 0,1 myF
1 330 pF

Widerstände:

2 0,33 K
1 0,39 K
1 1,8 K
1 3,6 K
2 9,1 K
3 10 K
3 12 K
1 47 K
1 82 K
2 220 K

Klaus Klöhn
Borchertring 83

2000 Hamburg 60

040/631 05 90

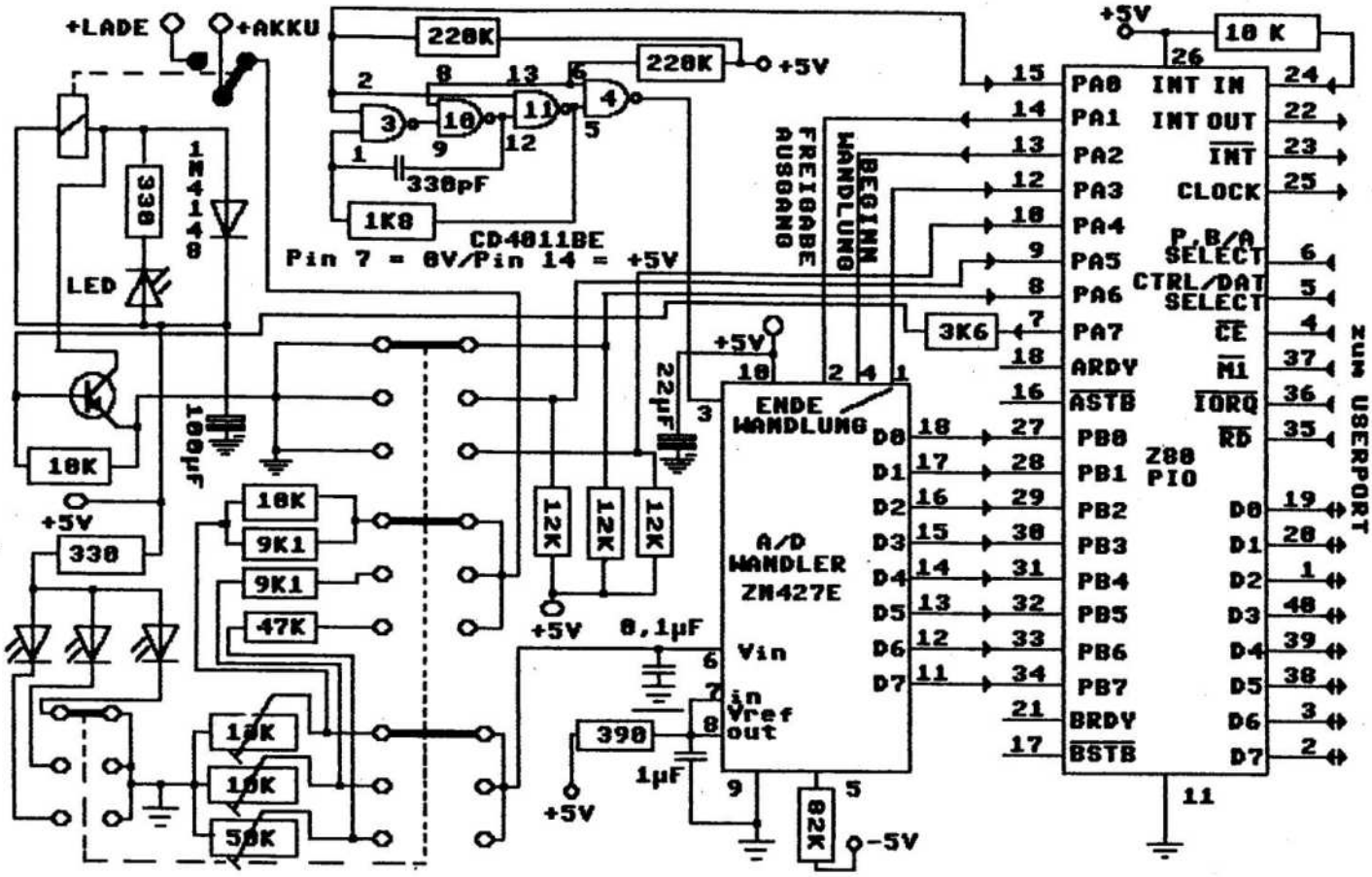
Ich habe außerdem folgendes zu verkaufen:

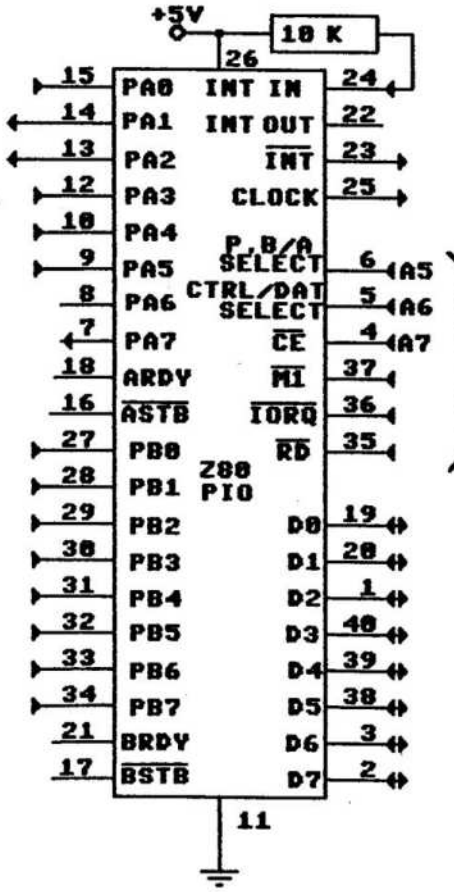
1	Spectrum plus	DM 150
1	Spectrum in Lo Profile Tastatur	DM 150
1	Interface 1	DM 80
3	Microdrive je DM 80	DM 240
1	Kempston E Drucker Interface	DM 60
1	Kempston S Drucker Interface	DM 30
1	Sinclair Drucker mit Papier	DM 50
50	Microdrive Cartridges mit Programmen je DM 5	DM 250
2	Kempston Joystick Interfaces je DM 20	DM 40
19	Original Spiele je DM 5	DM 95
6	Original Anwender Programme: z.B. Skyline Basic, Supercode, Datenmanager, Devpak	DM 60
80	Kassetten (C90 / C60) mit Programmen	im Paketpreis als Zugabe
12	Bücher je DM 5	DM 60
90	Zeitschriften	Zugabe
1	Spectrum 128 (def. ULA ?)	DM 50
1	ULA (48)	DM 50
1	ULA Microdrive	DM 20

alles zusammen für DM 500,-

und dann im Paket mit Abschaltautomatik für Akku-Schnellladegeräte
Teilekosten ca. DM 60,-

Spectrum User Club Wuppertal



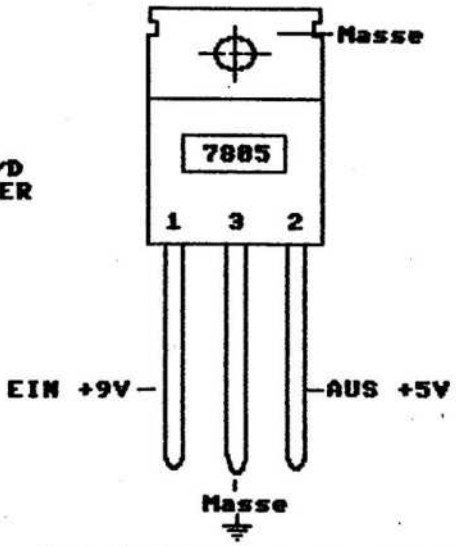


PRINTPLATTENSTECKER an PIO-BAUSTEIN

*	24B	24A	MI	
A5	23B	23A	*	
A6	22B	22A	*	
A7	21B	21A	*	
*	20B	20A	-5V	an A/D WANDLER
*	19B	19A	*	
*	18B	18A	RD	
*	17B	17A	IORQ	
*	16B	16A	*	
*	15B	15A	*	
*	14B	14A	*	
*	13B	13A	INT	OBERSEITE
*	12B	12A	D4	
*	11B	11A	D3	
*	10B	10A	D5	
*	9B	9A	D6	
CLOCK	8B	8A	D2	
0V	7B	7A	D1	
0V	6B	6A	D0	
SCHLITZ	5B	5A	SCHLITZ	
+9V	4B	4A	*	
*	3B	3A	D7	

UNTERSEITE

Die +5Volt Versorgung



Den Spannungsregler auf Kühlblech montieren

+9Volt Eingangsspannung stehen an 4B,
-5Volt für A/D-Wandler stehen an 28A
vom Printplattenstecker

```
 10 REM         *SCHNELLADEKONTROLLER* FUER NC-AKKUS
 20 REM              VON    K.-R. KLOEHN  HAMBURG
 30 REM
 40 REM
 50 POKE 23658,8
 60 LET DPA=31
 70 LET KPA=95
 80 LET DPB=63
 90 LET KPB=127
100 REM ..... DPA= DATEN PORT A. KPA= KONTROLLWORT PORT A. PORT B ENTSPRECHEND
110 REM ..................................... EIN-AUSGAENGE WERDEN FESTGELEGT
120 OUT KPA,255: REM ..................................... ( BIN,11111111 )
130 OUT KPA,120: REM ..................................... ( BIN,01111000 )
140 OUT KPB,255
150 OUT KPB,255
160 REM ..................................... MESSEINRICHTUNG AUF NULL SETZEN
170 OUT DPA,0
180 REM ..................................................... DIE VARIABLEN
190 LET MIN=0: REM ..................................... ZEITMESSER
200 LET NORM=5: REM ..................................... ANFANGSNORMIERZEIT
210 LET PAU=5: REM ..... ABTASTINTERVALL IN SEK. WIRD GEANDERT IN PAUSE ...
220 LET SCHALT=3: REM ..ANZAHLKLEINERER MESSUNGEN ( ZWEI WERTEKLEINER ALS MAX )
230 LET X=0: LET Y=0: LET PLOT=0: REM ..... PLOTPOSITION AUF ANFANG
240 LET ZAHL=0: REM ..... ANZAHL DER MESSUNGEN
250 LET MAX1=0: REM ..... MAXIMALER EINGANGSPEGEL PORT B
260 LET C=0: REM ..... ZAEHLER FUER ANZAHL GROESSERER MESSUNG
270 LET MERK=0: REM ..... MERKER FUER MAXPEGEL
280 LET MERKER2=0: REM ..... MERKER FUER KLEINERE SPANNUNG
290 LET G=0: REM ..... ZAEHLER FUER MERKER 2
300 LET VIER=0: REM ..... ZAEHLER FUER PLOTTEN
310 LET MESS=0: REM ..... EINGANGSPEGEL AM PORT B
320 LET B=0: REM ..... EINGANGSPEGEL AM PORT B
330 LET RE=0: REM ..... UMRECHNER FUER *MESS* IN VOLT
340 LET A=0: REM ..... SCHALTERSTELLUNG FUER AKKU-ANZAHL
350 LET ANZAHL=0: REM ..... ANZAHL AKKUS MAXIMAL IN AKTUELLER SCHALTERSTELLUNG
360 LET MAX=0: REM ..... RECHNER FUER AKTUELLE SPANNUNG
370 LET S=0: REM ..... SEKUNDEN
380 REM ..... EINSTELLMENUE
390 GO SUB 400: GO TO 460
400 BORDER 0: PAPER 0: INK 7: CLS
410 PLOT 0,0: DRAW 0,175: DRAW 255,0: DRAW 0,-175: DRAW -255,0
420 PLOT 0,144: DRAW 255,0
430 PLOT 0,143: DRAW 255,0
440 PRINT  INVERSE 1;AT 1,4;"**SCHNELLADEKONTROLLER**";AT 2,9;"FUER  NC-AKKUS"
450 RETURN
460 PRINT AT 5,4;"BITTE DEN SCHALTER FUER";AT 6,4;"MAXIMALE AKKU-ANZAHL";AT 7,4
;"EINSTELLEN"
470 PRINT AT 9,7; INVERSE 1;"SPACE"; INVERSE 0;" WENN FERTIG"
480 LET A=IN DPA: REM ..... ABFRAGE DER SCHALTERSTELLUNG UEBER PORT A
490 LET A=IN DPA: IF A=48 THEN  LET ANZAHL=4
500 IF A=80 THEN  LET ANZAHL=6
510 IF A=96 THEN  LET ANZAHL=10
520 REM .....AKKU-ANZAHL KANN BELIEBIG GEWAEHLT WERDEN. IST ABHAENGIG VON DE
530 REM .....EINSTELLUNG DER TRIMMER

540 PRINT AT 16,7; INVERSE 1;" MAXIMAL "; FLASH 1;ANZAHL; FLASH 0;" AKKUS "; IN
VERSE 0;" "
550 IF INKEY$=" " THEN  GO TO 570
560 GO TO 480
570 GO SUB 400
580 PRINT AT 16,7; INVERSE 1;" MAXIMAL "; FLASH 1;ANZAHL; FLASH 0;" AKKUS "; IN
VERSE 0;" "
590 PRINT AT 6,2;"DIE ANZAHL DER AKKUS KANN";AT 7,2;"WAEHREND DES LADEVORGANGES
";AT 8,2;"NICHT MEHR VERAENDERT WERDEN"
600 PRINT AT 10,4;"EINSTELLUNG RICHTIG ?";AT 11,10; INVERSE 1;"J"; INVERSE 0;"A
/ "; INVERSE 1;"N"; INVERSE 0;"EIN"
610 IF INKEY$="N" THEN  GO TO 390
620 IF INKEY$="J" THEN  GO TO 640
630 GO TO 610
640 GO SUB 400
650 PRINT AT 5,1;"FOLGENDE WERTE SIND VORGEGEBEN"
660 PRINT AT 8,1; INVERSE 1;"ANFANGSNORMIERZEIT "; INVERSE 0;"   ";NORM;"  MIN"
670 PRINT AT 10,1; INVERSE 1;"GLEICHE MESSUNGEN  "; INVERSE 0;AT 11,1;"ZUM ABSC
HALTEN";AT 10,23;SCHALT
680 PRINT AT 12,1; INVERSE 1;"ABTASTINTERVALL    "; INVERSE 0;"   ";PAU;"  SEK"
690 PRINT AT 15,1;"WUENSCHEN SIE ANDERE WERTE ?";AT 16,9; INVERSE 1;"J"; INVERS
E 0;"A / "; INVERSE 1;"N"; INVERSE 0;"EIN"
700 IF INKEY$="J" THEN  GO TO 730
710 IF INKEY$="N" THEN  GO TO 900
720 GO TO 700
730 PRINT AT 15,1;" BITTE EIGENEN WERT EINGEBEN  ";AT 16,9;"           "
740 FOR F=22785 TO 22785+18: POKE F,242: NEXT F
750 INPUT "ANFANGSNORMIERZEIT: ";NORM
760 PRINT AT 8,23;NORM
770 FOR F=22785 TO 22785+18: POKE F,7: NEXT F
```

```
 780>FOR F=22849TO 22849+18:POKE F,242:NEXT F
 790 INPUT "GLEICHE MESSUNGEN: ";SCHALT
 800 PRINT AT 10,23;SCHALT
 810 FOR F=22849 TO 22849+18: POKE F,7: NEXT F
 820 FOR F=22913 TO 22913+18: POKE F,242: NEXT F
 830 INPUT "ABTASTINTERVALL: ";PAU
 840 PRINT AT 12,23;PAU
 850 FOR F=22913 TO 22913+18: POKE F,7: NEXT F
 860 PRINT AT 15,1;"      FERTIG ?    "; INVERSE 1;"J"; INVERSE 0;"A / "; INVERSE
 1;"N"; INVERSE 0;"EIN   "
 870 IF INKEY$="J" THEN  GO TO 900
 880 IF INKEY$="N" THEN  GO TO 640
 890 GO TO 870
 900 LET PAU=(PAU*50)
 910 GO SUB 400
 920 PRINT AT 6,1;"AKKUS/LADEGERAET ANSCHLIESSEN";AT 8,4;" DANN "; INVERSE 1;"EN
TER"; INVERSE 0;" DRUECKEN"
 930 PRINT AT 20,8; INVERSE 1;"MAXIMAL "; FLASH 1;ANZAHL; FLASH 0;" AKKUS"; INVE
RSE 0
 940 PAUSE 0: PRINT AT 8,4;"                                "
 950 PRINT AT 11,11; FLASH 1;"AKKUTEST"; FLASH 0
 960 PRINT AT 12,9;"ETWAS GEDULD"
 970 FOR F=1 TO 10: REM .....................................ZEHN MESSUNGEN
 980 OUT DPA,0: REM ..................................... ALLES AUF NULL
 990 OUT DPA,BIN 00000111: REM ..... AD-WANDLER FREIGEBEN UND TAKT STARTEN
1000 FOR K=1 TO 10: NEXT K: REM ................. KURZE PAUSE FUER DEN WANDLER
1010 LET MESS=IN DPB: LET B=MESS: REM ........EINGANGSPEGEL AM WANDLER ABFRAGEN
1020 PRINT AT 14,1;"IN PORT B       ";MESS
1030 LET RE=MESS: GO SUB 1630: LET MESS=RE: REM .............UMWANDLUNG IN VOLT
1040 PRINT AT 16,1;"AKKUSPANNUNG    ";MESS;" VOLT    "
1050 NEXT F
1060 OUT DPA,0: OUT DPA,BIN 10000000: REM .................LADEGERAET EINSCHALTEN
1070 PAUSE 1000: OUT DPA,0: FOR F=1 TO 10: NEXT F: REM .KURZE PAUSE FUERS RELAIS
1080 OUT DPA,BIN 00000111
1090 LET MESS=IN DPB: LET B1=MESS: PRINT AT 14,16;B1: LET RE=MESS: GO SUB 1630:
LET MESS=RE
1100 PRINT AT 16,16;MESS: PAUSE 100
1110 OUT DPA,0: IF B1<=B THEN  GO SUB 400: PRINT AT 6,4;"LADEGERAET UEBERPRUEFEN
": LET MAX=0: LET MIN=0: PAUSE 200: GO TO 910
1120 OUT DPA,0: IF B=0 AND B1=0 THEN  GO SUB 400: PRINT AT 8,3;"AKKUS UEBERPRUEF
EN": PAUSE 200: GO TO 910
1130 CLS
1140 PRINT #0; INVERSE 1; BRIGHT 1;"     SCHNELLADEKONTROLLER        "
1150 PRINT #1; INVERSE 1;"          FUER NC-AKKUS          "
1160 PLOT 0,0: DRAW 0,175: DRAW 255,0: DRAW 0,-175: DRAW -255,0: PLOT 0,38: DRAW
 255,0: PLOT 0,39: DRAW 255,0
1170 REM ..........................................................HAUPTMENUE
1180 PRINT  INVERSE 1;AT 18,1;"ANFANGSNORMIERZEIT                "
1190 PRINT  INVERSE 1;AT 20,1;"AUTOMATIK AUF LADEN GESCHALTET"
  1200 POKE 23672,0: POKE 23673,0: POKE 23674,0: REM ..............UHR AUF NULL
1210 OUT DPA,0: REM ..................................................ALLES AUF NULL
1220 FOR f=1 TO 10: NEXT f: REM .....................KURZE PAUSE FUER DAS RELAIS
1230 OUT DPA,BIN 00000111: REM ...........TAKT STARTEN UND AD-WANDLER FREIGEBEN
1240 LET MESS=IN DPB: LET B=MESS: REM UMWANDLUNG IST ERFOLGT.( BEI MC UMWANDLUNG
 ABFRAGEN ).MESS UND B = AKTUELLE EINGANGSSPANNUNGEN
 1250 OUT DPA,BIN 10000000: REM .....................RELAIS AUF LADEN SCHALTEN
1260 PRINT AT 5,1;"IN/ PORT B";AT 5,22;MESS;"    "
 1270 REM ..........................................ANZEIGE DER AKTUELLEN MESSUNG
1280 LET ZAHL=ZAHL+1: PRINT AT 7,1;"ANZAHL MESSUNGEN";AT 7,22;ZAHL: REM ANZAHL D
ER MESSUNGEN
1290 REM ............UMRECHNUNG DER MESSWERTE IN VOLT. ANZAHL = ANZAHL DER MAX A
KKUS. SCHALTERSTELLUNG WIRD IM PROGRAMM UEBER PORT-A ABGEFRAGT.
1300 LET RE=MESS
1310 GO SUB 1630
1320 LET MESS=RE
1330 PRINT AT 3,1;"AKTUELLE SPANNUNG";AT 3,22;MESS;" V   "
 1340 REM ..........................................ANZEIGE DER SPANNUNG IN VOLT
1350 IF MESS=0 THEN  LET B=0: LET B1=0: GO TO 1120: REM ..........NOTABSCHALTUNG
1360 IF NORM-MIN>0 THEN  PRINT AT 18,24;NORM-MIN; INVERSE 1;" MIN": PAUSE PAU: G
O TO 1560
1370 REM ..............BERECHNUNG DER NEUEN MAXIMALSPANNUNG ( DREI MESSUNGEN )
1380 IF B>MAX THEN  LET C=C+1: LET MERK=MERK+B: IF C>2 THEN  LET MAX=INT (MERK/3
): LET C=0: LET  MERK=0
1390 IF B<=MAX THEN  LET C=0: LET MERK=0
1400 LET RE=MAX: GO SUB 1630
1410 LET MAX1=RE
1420 PRINT AT 1,1;"MAXIMALSPANNUNG";AT 1,22;MAX1;" V    "
1430 IF NORM>MIN THEN  PRINT AT 18,1; INVERSE 1;"ANFANGSNORMIERZEIT NOCH ";NORM-
MIN;" MIN";AT 20,1;"AUTOMATIK AUF LADEN GESCHALTET"
1440 REM ............BERECHNUNG OB SPANNUNG FAELLT. BEI 2 PUNKTEN UNTER MAX-WER
T (ANZAHL MESSUNGEN IST EINGESTELLT) IST AKKU VOLL. ABSCHALTEN.
 1450 REM .......................ABSCHALTUN ERFOLGT ERST NACH ANFANGSNORMIERUNG
1460 IF MAX-B>10 THEN  LET B1=0: LET NORM=1: GO TO 1110
1470 IF B<MAX THEN  LET G=G+1: IF MAX-B>1 THEN  LET MERKER2=MERKER2+1: IF MERKER
2>(SCHALT-1) THEN  BEEP 1,10: OUT DPA,0: PAUSE 0: RUN : REM ...AKKU IST VOLL
1480 IF B>=MAX THEN  LET MERKER2=0: LET G=0
1490 IF MAX-B<2 THEN  LET MERKER2=0
1500 PRINT AT 14,1;"KLEINERE MESSWERTE = ";MERKER2
```

```
1510>REM ES FOLGT GRAFISCHE DARSTELLUNG DES VERLAUFES DER AKKUSPANNUNG.( NACH AN
FANGSNORMIERUNG )
1520 IF PLOT=0 THEN  LET PLOT=B: PRINT AT 18,1;"                              ";
AT 20,1;"                                                                     "
1530 LET VIER=VIER+1: IF VIER>INT (((3600/(PAU/50))/240)+1) THEN  LET VIER=0: GO
 TO 1550: REM .....................RECHNER WIE OFT GEPLOTTET WIRD
1540 GO TO 1560
1550 LET X=X+1: LET Y=(B+5)-PLOT: PLOT x+2,y: PAUSE PAU: GO TO 1210: REM .......
.............RUECKSPRUNG NACH DARSTELLUNG DES SPANNUNGSVERLAUFES
1560 LET S=(PEEK 23672+256*PEEK 23673)/50: IF S>60 THEN  POKE 23672,0: POKE 2367
3,0: POKE 23674,0: LET MIN=MIN+1: REM ........ UHR WIRD GESTELLT
1570 PRINT AT 9,1;"LADEZEIT";AT 9,22;MIN;" MIN": PAUSE PAU: GO TO 1210: REM ....
.............. RUECKSPRUNG ZUR MESSSCHLEIFE NACH KURZER LADEZEIT
1580 PRINT AT 0,0;"PEEK 23672 =   ";PEEK 23672
1590 PRINT AT 2,0;"PEEK 23673 =   ";PEEK 23673
1600 PRINT AT 4,0;"PEEK 23674 =   ";PEEK 23674
1610 GO TO 1580
1620 REM .....................UMRECHNUNG IN VOLT
1630 IF anzahl=4 THEN  LET RE=RE*0.0313725
1640 IF anzahl=6 THEN  LET RE=RE*0.0470588
1650 IF anzahl=10 THEN  LET RE=RE*0.0784313
1660 LET RE=RE*100
1670 LET RE=INT RE
1680 LET RE=RE/100
1690 RETURN
```

****

LISTING 2

```
  30 OUT 95,255: OUT 95,120: OUT 127,255: OUT 127,255
  40 OUT 31,BIN 10000000
  50 PRINT AT 10,0;"DAS RELAIS SCHALTET EIN"
  60 FOR F=1 TO 500: NEXT F
  70 OUT 31,0
  80 PRINT AT 10,0;"DAS RELAIS SCHALTET      AUS"
  90 FOR F=1 TO 500: NEXT F
 130 OUT 31,0: OUT 31,BIN 00000111
 140 FOR f=1 TO 10: NEXT f
 150 PRINT AT 12,0;IN 63;"  ";"IN PORT B  "
 160 LET A=IN 31
 170 IF A=63  THEN  LET B=4: LET C=0.0313725
 180 IF A=95  THEN  LET B=6: LET C=0.0470588
 190 IF A=111  THEN  LET B=10: LET C=0.0784313
 200 PRINT AT 10,0;B;" AKKUS MAX                          "
 210 LET V=IN 63*C: LET V=V*100: LET V=INT V: LET VOLT=V/100
 220 PRINT AT 15,0;VOLT;" VOLT AKKUSPANNUNG  "
 300 GO TO 130
```

Karl-Heinz Meander, Am Schloßberg 19, 8431 Hohenfels

Hallo Club- und Spectrumfreunde,

fast Zeitgleich mit dem Infoheft 12/1989, erhielt ich die Beta Basic Newsletter 8-14. Interessant war der Beitrag von Manfred Nolten über den "Spectrum clone" SAM.

Zu diesem Projekt SAM hat auch der Autor von Beta Basic und zugleich der Autor des SAM Basic, Andy Wright einen Artikel in seinen "Newsletter" veröffentlicht, dem ich keinen der Clubfreunde vorenthalten möchte.

Ich habe bewußt auf eine Übersetzung verzichtet, damit sich nicht irgendein Übersetzungsfehler einschleicht. Ich glaube daß sicher ein jeder soviel Englisch beherrscht um den Beitrag zu verstehen.

Ich zitiere Andy Wright:

SAM - THE NEW MICRO FROM MGT

Quite a few readers have noticed stories in the computer press revealing that I am writing the Basic for the new SAM micro, and asked for details. In fact, a number of readers suggested my name to MGT in the first place - thanks for the vote of confidence! I cannot reveal some things, partly because they are confidential, and partly because production machines aren't ready yet and some details are not fixed. However, that still gives me plenty of scope.

The machine has been called a "Spectrum clone" wich is a slur on a machine which completely outclasses the Spectrum. It is true that one screen mode is like the Spectrum's, and that the machine is flexible enough to run most 48K games. (You would be able to run even more software if you loaded a tape copy of the Spectrum's ROM into SAM's paged RAM.) It might be true to say that SAM is to the Spectrum what the Spectrum was to the ZX81. Certainly my idea of what a Basic should be has been influenced by ZX Basic - I happen to like syntax check on entry, string "slicers", automatic listings, etc. Clive Sinclair should have developed a machine like this, instead of the 128K Plus. Towards the end, he didn't seem to employ any imaginative engineers, though, and it has been left to a small, ictive outfit like MGT to fill the gap.

SAM has a Z80B "brain" chip, wich runs 70% faster than the Z80A in the Spectrum. (If I write a particular Basic command no better than the Spectrum ROM version, I still get a 70% speed increase.) You get 256K or 512K of RAM, and further expansion is possible via the edge connector. The RAM is very cleverly paged to make life easy for machine-code types like me; Basic users will simply notice the nice features it allows me to provide, like BIG programs and arrays (100's of kilobytes), instantly switchable screens, fast operation, etc.

There are 4 screen modes. One major improvement in all modes is that the 16 colours we are used to on the Spectrum (8 bright, 8 non-bright) actually select an entry in a special palette memory; the contents of that memory then detrmine the clour you see on the screen. The 16 palette memories can be programmed with any of 64 different colours, mixed from 4 intensity levels for blue, red and green (4*4*4=64). With appropriate values, a game or screen picture will look the same as it did on the Spectrum, but often you would want to completely alter the colours. Even better, you can change the palette memory contents part-way down a screen, at any position, multiple times! This allows all 64 colours to be used at once, with some restrictions on location. Even the border colour can change part-way down a screen.

Colours can also be made to flash - e.g. COLOUR 173d,45 would make colour 1 (normally blue on the Spectrum) flash between colour values 30 and 45. (I am not sure what actual names we will use for the colours yet - "red with a touch of blue and a dash of green?".) This can give pixel-level FLASH.

MODE 1: Just like the Spectrum's screen, using 6K for "Pattern" data and 0.75K for attributes. Attribute clashing as usual!

MODE 2: This uses 6K for "Pattern" data and 6K for attributes. Resolution is Spectrum's 256*192 pixels, but any 8*1 pixel row in a character square can have its own Spectrum-style attributes.

MODE 3: Uses 24K. There is no separate block of attributes. Instead, each pixel has its own, as part of the main screen memory. Mode 3 has a resolution of 512*192 pixels, and allows 4 colours to be used at once from the palette of 16. (Which can be altered part-way down the screen.) You can have up to 856-pixel characters across the screen. I suppose this mode will be mainly used for word-processing and other detailed work. The use of any 4 colours is pretty good, and you can simulate more with stippled FILLS. However, my favourite is the next mode - mode 4.

MODE 4: Uses 24K. Resolution is 256*192, with 16 possible colours per pixel. It is very nice to get away from attribute problems! I recently ran a Mandelbrot set program to give complex, detailed colour patterns that would be quite impossible on the Spectrum.

Modes 2-4 all have nice easy memory arrangements for machine-code types - e.g. mode 4 uses 192 contiguous rows of 128 bytes, each half-byte coding for a pixel. To get a screen address from x and y coordinates takes 3 machine-code instructions, vs. 20 or more on the Spectrum. This partly explains why e.g. DRAW is about 7 times faster on the SAM than on the Spectrum. (Other reasons are the faster processor, and the fact that I thought very hard when writing the graphics commands!) Using half-bytes for each pixel is a good idea because it makes left and right ROLL and SCROLL by one pixel easily possible with the Z80's half-byte rotate instructions.

SAM's sound chip is better than the on provided on the 128K Spectrum, but I am not doing the programming on that part of the project, so I can't comment further on the sound facilities.

The keyboard will be "proper", with a set of function keys, and completely "soft" - that is, the keys will be programmable so they can produce any character you want. This, combined with an extensive international character set, should make the machine particularly attractive in overseas markets.

SAM will be compatible with most Spectrum software, so it will start with a big software base, unlike most new machines. SAM Basic will do everything that Spectrum Basic can do, so Basic programs that don't try to be too clever with PEEKs and POKEs and USRs should run. A small and painless translation step is all that is required. As I write, I am not sure how many of Beta Basic's features will be packed into SAM's 32K ROM - all the majors commands should be there, though. Plus others to support SAM's hardware. What about compatibility with existing BB programs? Well, I cannot be completely tied to the current way of doing things, wich are in some ways a "kludge" forced by the Spectrum's ROM. For example, using line zero to provide extra functions, and using UDGs as new keywords, will have to be ditched. But there are so many points of compatibility that you will normaly be able to LOAD a Beta Basic program and then use some kind of SAM Basic translator program (full of ALTERs, I expect) to make the program runnable.

The machine should be available from April 1989; the price will be about £140.00 for the basic 256K version. A compatible disc system will be available. MGT are allowing interested customers to reserve a place in the queue for machines at no charge- write to them if you are interested in doing this. Their address is:

MGT, LAKESIDE, PHOENIX WAY,
SWANSEA ENTERPRISE PARK, SWANSEA, SA7 9EH

Das war also der im Frühjahr dieses Jahres veröffentlichte Bericht von Andy Wright.

Wie man an dem Beitrag von Manfred Nolten erkennen kann ist der Preis ein etwas höherer als von Andy Wright angegeben und natürlich stimmt das Datum des Erscheinens nicht. Jedoch hat man dadurch, daß Dr. Wright der schöpfer und schreiber des SAM Basic ist, informationen aus erster Hand erhalten. Interessant ist auch, daß es sogar möglich ist in Beta Basic geschriebene Programme in den SAM zu laden, wenn man auch wie Andy erwähnt einiges mit ALTER anpassen muß.

Liebe Spectrum-Freunde,

es wird wieder interessant! Der so lange angekündigte SAM ist nun endlich da. Beim Erscheinen dieses Info's dürften die ersten Geräte bereits hier eingetroffen sein. Obwohl sich die ganze Geschichte über mehrere Monate hingezogen hat, hat sich die Warterei mehr als gelohnt. Nach eingehender Lektüre des technischen Handbuches für Programmierer kann man mit Fug und Recht behaupten, daß diese Maschine keinerlei Wünsche offenläßt. Jeder "Otto Normalverbraucher" und natürlich auch die MC-Freaks findet an diesem Computer etwas, was er bis jetzt vielleicht vermißt hat. An dieser Stelle sollen nun kurz die Befehle des "SAM-BASIC" vorgestellt werden. Ausführlicheres findet sich im o.g. Programmierer-Manual (englisch). Die Beschreibung aller Möglichkeiten würde diesen Rahmen sprengen. Das Manual ist bei mir vorab zum Preis von 25,--DM erhältlich.

SAM-BASIC (Auszug aus rund 160 Befehlen und mehr...)

PEN (wie INK)
INVERSE
OVER (0,1,2,3)
KEY posn,x (CODE-Zuordnung der Tastatur)
DEF KEYCODE (Belegung der Tasten mit z.B. Strings oder Befehlsfolgen)
LPRINT, PRINT, DEF FN (Funktionsnamen beliebiger Länge)
BLOCKS (Block-Grfik), RECORD TO a$ (Schreibt Grafik-Befehlsfolgen in einen String; erhebliche Steigerung der Grafikgeschwindigkeit beim Abrufen der Befehlsfolge mit BLITZ a$), RECORD STOP, BLITZ a$,
CLOSE, OPEN (INPUT#2; x ist jetzt möglich für Masken ohne Bildschirmlöschen), DEVICE (legt Ein-/Ausgabegerät fest), AUTO, BOOT,
BORDER 0-15, CALL adr,x,y,fred$, CLEAR n (n=90000 ist erlaubt),
DO, WHILE, UNTIL, LOOP, WHILE, UNTIL, EXIT IF, LOOP IF, DRAW,
DUMP (Screen copy), DUMP CHR$ (text copy), GET, FATPIX, FILL (arbeitet mit allen Grafik-Mustern), GRA$ a$ (speichert Bildschirminhalte oder Teile davon in einen String), PUT x,y,a$ (holt diese wieder zurück an beliebige Stelle), LABEL (z.B. 2ØØ LABEL Hölle:PRINT "Das ist die Hölle"
1Ø GOTO Hölle Ergebnis: Das ist die Hölle), DEFAULT, LIST FORMAT f,
ON ERROR, POKE, (POKE adr,234,54,0,77,255,4,255 ist möglich), DPOKE
TPOKE (Doppel- oder dreifach-POKE), POP, DEF PROC, LOCAL, RENUM, ROLL,
SCROLL, SOUND, OPEN SCREEN n,m (Reservierung von bis zu 16 SCREENs,
die dann mit DISPLAY n abgerufen werden können), WINDOW, BIN$, HEX$,
und vieles, vieles andere mehr.....

AUSTEL, HOFHEIMER- 53, 6238 HOFHEIM 7, TELEFON (06192)27654 (18-20h)

Werner Nuhn
Eichstachelstr. 16
6301 Allendorf/Climbach

Climbach, 17.01.90

"Suche Beta-Disk-Interface für
ZX-Spectrum 48 K, Version 4/12 o. ä."

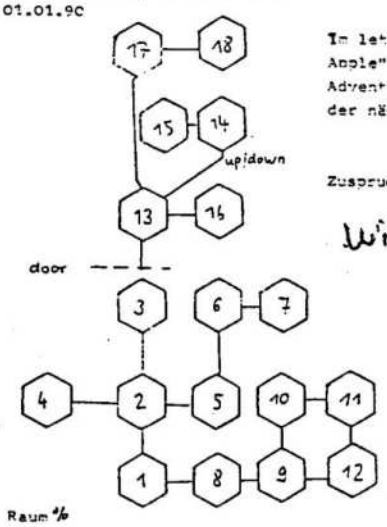
Text-o-Mat für Tasword II

Falls Interesse besteht werde ich in loser Folge einzelne Teile meines Textautomaten beschreiben, den ich mir in jahrelanger Ar-Arbeit zusammengebastelt habe. Jeder kann davon für sein Textsystem das entnehmen, was er für seine Zwecke brauchen kann, denn alles geht nicht in ein Basicprogramm hinein. Man kann dieses aber teilen oder drei und mehr Spezialversionen anfertigen und diese nach Bedarf nachladen (andere Programme machen das ja auch). Ich versuche damit zugleich zu zeigen, wie man auf ganz unkonventionelle Art programmieren kann, aber auch, wie man die verschiedensten Programmierstile zusammenbasteln kann.
Als erstes gilt es, das ursprüngliche Tasword Basic radikal abzuspecken. Wenn man alle Zahlen durch VAL- Werte ersetzt, kann man schon etwa 1100 bytes sparen. Dann kann man 0 durch NOT PI, 1 durch SGN PI, 2 durch SQR PI, 3 durch PI, 5 durch SQR EXP PI, 6 durch EXP SQR PI,10 durch PI*PI, 23 durch EXP PI ersetzen. Das spart wieder jeweils ein bis 2 bytes, die gebrochenen Zahlen gehen aber nicht bei Grafik- oder Rechenanwendungen. Bei hohen Zahlen bringt z.B. VAL "3e4" nochmal bytes gegenüber 3e4.
Dann werden die aufwendigen Menues die man nach Einarbeitung ja nicht mehr braucht, entschlackt. Hier ist wieder einiges zu holen, die Worte gekürzt, jeder Buchstabe spart 8 bytes, PRINT AT nach Möglichkeit durch PRINT ersetzt, einzelne Zeilen zusammengelegt. So können vom ursprünglichen Basic etwa 5470 bytes eingespart werden.
Platz für die vielen zusätzlichen Maschinenroutinen werden wir auf den Helpseiten finden, die wir ja nur zur Einübung des Tasword gebraucht haben. Beschränkt und mit Vorsicht auch noch im Maschinencode des Tasword selbst und wir können selbst Teile von diesem, z.B. ab 64000 durch Nachladeroutinen überschreiben und damit arbeiten, wenn wir nur rechtzeitig, bevor wir in den Tasword Editor zurückgehen, diese wieder reparieren. So können wir unseren bescheidenen Speicherplatz und die Möglichkeiten von Tasword in nicht für möglich gehaltenem Ausmaß steigern, soweit, das man Tasword selbst kaum noch als Hauptprogramm ansehen kann. Im Prinzip wäre es ersetzbar, doch kenne ich kein Programm, das eine solche Möglichkeit bietet. Alle neueren Programme haben zu wenig Platz für zusätzliches Basic und man hat somit nicht die Möglichkeit, sich ein Programm zuzuschneiden, das den eigenen Erfordernissen 100% entspricht, das jeden Wunsch erfüllt.
Die anderen Programme sind Konfektion, dieses Programm ist ein Maßanzug!
Beginnen wir also mit dem Abspecken, der Frühjahrskur für unser Programm. Wer noch die Originalfassung hat, findet unten dazu ein Listing. Das ursprüngliche Program hat 6735 bytes, das neue 1165 bytes. Wer schon Zusatzprogramme eingebaut und seinen Drucker weitgehend angepaßt hat, muß das entsprechend selbst besorgen. Für die Umstellung auf VAL- Werte gibt es Programme in Beta Basic und auch MC- Routinen, die das für uns machen. Wer eine Discovery- Diskettenstation hat, tut sich mit dem Löschen vieler Zeilen leicht und mit dem Herauslösen von Einzelroutinen aus einem Programm: CLEAR LINE (erste Zeile),(letzte Zeile) löscht die angegebenen Zeilen.
Die alte englische Version habe ich vorher mit einer VAl- Routine auf VAL- Werte ändern lassen. Ich habe dabei schon 1165 bytes gewonnen. Ich arbeite jetzt sozusagen LIVE mit meinem Textautomat. Dieser Text ist im Editor. Ich gebe nun über STOP-b- NEW ein. Dann merge ich das obige Basic. Mit RUN bin ich wieder in diesem Text. Jetzt habe ich natürlich alle meine neuen Möglichkeiten nicht mehr, aber ich kann das Programm doch bearbeiten und immer wieder abspeichern und kann mir immer ansehn, wieviel bytes ich gewonnen habe. Dazu setze ich auf Zeile 9900 SAVE *1;"Tas" und auf Zeile 8094 PRINT VAL "65535"- USR VAL "7962": PAUSE PI: PAUSE NOT PI: GO TO PI. Diese Möglichkeit habe

ich durch Zusatzroutinen im erweiterten MC. (Erklärung später!) Ich bekomme 2485 freie Speicherbytes. In Zeile 10 habe ich nun GO TO VAL "20" durch GO TO EXP PI ersetzt und wieder 3 bytes gespart. Natürlich muß ich dann aus Zeile 20, Zeile 23 machen. Zeile 11 kann ich ganz löschen (alberne Piepserei!) und gewinne 46 bytes, beim Säubern der Zeile 15 103 bytes. Zeile 25-190 löschen wir zunächst und gewinnen 700 bytes. Wir sind jetzt schon auf 3362 bytes freier Basicspeicherplatz, das Programm hat jetzt 4767 bytes. Immer wieder abspeichern nicht vergessen, sonst ist die Arbeit irgendwann umsonst gewesen.
Das Programm ist im Moment nicht lauffähig. Wir geben in Zeile 27 eine Steuerung ein, die bei Speicherplatz 23560 die gedrückte Taste abfragt und ändern die Zeilen der einzelnen Menuepunkte sinngemäß. j= 106*10 = 1060, also kommt die Laderoutine auf 1060. Dabei fällt wieder viel Ballast vom Originalprogramm ab.
Den Menuepunkt- g, Definition der Druckersteurzeichen streichen wir ganz und bringen eine bequemere Eingabemöglichkeit später.
Ebenso lassen wir die ganze REPLACE- Routine weg. Wir bekommen etwas besseres dafür und der Aufruf EXT-r kann viel besser genützt werden, weil wir nur von hier aus den Stand des Cursors berechnen und manipulieren können.
Nach allen Änderungen haben wir jetzt 6818 bytes freien Speicherplatz gewonnen und die Programmlänge ist auf 1265 geschmolzen. Dieses Programm sollten wir uns aufheben, wir können es immer wieder für neue Zusammenstellungen verwenden. Davon das nächste Mal.

Herbert Hartig
Postfach 147,8938 Buchloe

Haselsteig 41, 1000 Berlin 47
01.01.90



Im letzten ARTIC-Adventure "Golden Apple" (die Lösung des vorletzten Adventures "Inca Curse " folgt in der nächsten Ausgabe) sitze ich fest.

Zuspruch und Unterstützung hoch willkommen !

Raum %

1 road near mansion
2 mansion grounds
3 front porch, locked door
4 gate house
5 flower garden
6 vegetable plot
7 greenhouse
8 small country road
9 in wood a
10 in wood b
11 in wood c
12 in wood d
13 hallway
14 landing
15 bedroom
16 study room
17 living room
18 dining room

LÖSUNGSWEG :

%4 : get box of matches
%5 : smell flowers ; get handkerchief; get key
%7 : get Can; get seeds
%10: fill Can
%11: climb tree ; get axe; down
%12: break log ; get planks; get /silver bar/;
%6 : plant seed; water seeds look; get carrots
%3 : unlock door; go door
%17: examine painting;look; get /horloge/
%18: climb table ; ?
%13: climb stairs
%15: examine dust;look; get /+RUG+/ push bed; look; get metal whistle
%16: leave all

Michael

Frank-Michael MOCZKO
Haselsteig 41, 1000 Berlin 47

```
 10 CLS : LET y=SGN PI: LET z=NOT PI: LET c=z: LET a=USR VAL "64330": GO TO EXP
 PI
 12 DEF FN p(x)=PEEK x+VAL "256"*PEEK (x+SGN PI)
 15 POKE VAL "23609",SQR PI: CLEAR VAL "31999": LOAD *SGN PI;"tasword"CODE : CL
S : LET a=USR VAL "59081": GO TO PI
 23 CLS : LET a=VAL "64"*INT (a/VAL "64"+VAL "0.99"): IF a=NOT PI THEN GO TO VA
L "3e3"
 25 PRINT BRIGHT SGN PI;" jladen-mergen- yzum Text basic- speich.gSerien-tProg.
savepDruck"
 27 PAUSE NOT PI: IF PEEK VAL "23560"<VAL "98" OR PEEK VAL "23560">VAL "121" TH
EN GO TO EXP PI
 29 GO TO PEEK VAL "23560"*VAL "10"
210 INPUT "Zeilenabstand(1)";b$: IF b$="" THEN LET b$="1"
215 POKE VAL "62235",VAL b$
220 INPUT "Anfang Z.? (1)";b$: IF b$="" THEN LET b$="1"
230 LET c=VAL "64"*(INT VAL b$-SGN PI): LET b=c+FN p(VAL "62216"): LET x=VAL "6
0045": GO SUB VAL "950"
240 INPUT "Ende Z.? (last)";b$: IF b$="" THEN LET b=a-c: GO TO VAL "250"
245 LET b=VAL "64"*INT VAL b$-c
250 RANDOMIZE USR VAL "59806": RANDOMIZE USR (FN p(VAL "62472"))
260 CLS : PRINT "q beendet Druck!"
270 LET x=VAL "60049": GO SUB VAL "950"
275 LET c=PEEK VAL "62470": IF c<>NOT PI THEN LPRINT CHR$ c
280 RANDOMIZE USR VAL "60038"
285 LET c=PEEK VAL "62471": IF c<>NOT PI THEN LPRINT CHR$ c
290 RANDOMIZE USR VAL "59806": GO TO PI*PI
950 POKE x,b-VAL "256"*INT (b/VAL "256"): POKE (x+SGN PI),INT (b/VAL "256"): RE
TURN
980 STOP
1060 INPUT "Vorsicht! LOAD c";c$: IF c$="c" THEN GO TO PI
1062 LET a=USR VAL "59081"
1090 INPUT "Name:";a$: LET b=FN p(VAL "62216"): LOAD *SGN PI;a$CODE (a+b),((FN p
(VAL "62221")+VAL "22")*VAL "64"-a): GO TO PI
1120 GO TO VAL "2e2"
1150 LET b=FN p(VAL "62216"): INPUT "Name SAVE ";a$: SAVE *SGN PI;a$CODE b,a: VE
RIFY *SGN PI;a$CODE : PRINT ''"Text>> ";a$;"<< O.K"
1152 PRINT ''a;" bytes",a/PEEK VAL "62237";" Zeilen": PAUSE NOT PI: GO TO PI
1160 CLS : LET b$="tasword": GO SUB VAL "9900": SAVE *SGN PI;b$CODE VAL "5479",
VAL "10751": VERIFY *SGN PI;a$CODE : PRINT "O.K.": PAUSE EXP PI: GO TO PI
1210 GO TO PI
8094 PRINT VAL "65535"-USR VAL "7962": PAUSE PI: PAUSE NOT PI: GO TO PI
9900 SAVE *SGN PI;"Tas"
```

TASWORD II, abgespeckt v.H.Hartig, Postfach 147, 8938 Buchloe

---

DRINGEND!!! Ich suche eine Original - Kassette mit einem Datenbankprogramm für den Spectrum 48 K, mit dem ich unbegrenzt viele Datensätze auf Kassette abspeichern kann. H. Schwinning, Henkelshof 1-3, 5630 Remscheid 11, Tel. 02191-660192

Karl-Heinz Meander, Am Schloßberg 19, 8431 Hohenfels

Manchmal steht man vor dem Problem, daß eine Bilddarstellung, ein Text, nur teilweise gelöscht bzw. verändert werden soll. Brauchbar wäre da ein Teil-CLS Befehl.

Beim ZX81 ließ sich ein solcher Befehl in M-CODE ja schreiben und sogar hervorragend einsetzen. Das ganze sah dann etwa so aus:

```
LD    B,5
CALL  #0A2A
RET
```

Dieses kurze Programm löscht beim ZX81 beispielsweise die unteren fünf Bildschirmzeilen.

Beim Spectrum führt also nur das Positionieren mit PRINT AT zu einem ähnlichen Effekt.

Das hier nun vorgestellte Maschinenprogramm erledigt die gleiche Aufgabe schnell - es wirkt praktisch wie CLS, nur daß es zeilenweise löscht. Ein Blick auf das das Display-File des Spectrum hilft zunächst, die Problematik deutlich zu machen.

Der Bildspeicher ist in zwei Hauptblöcke unterteilt. der erste Block bietet für jeden auf den Bildschirm darstellbaren Punkt 1 Bit Speicherkapazität. Da vom Spectrum jedes Zeichen durch 8 Byte dargestellt wird, ergibt das bei 24 Zeilen und 32 Spalten einen Speicherbedarf von 6 KByte. Dieser Block beginnt wie bekannt mit der Adresse 16384 und endet mit Adresse 22527.

Nun wäre es naheliegend, daß die Hell-/Dunkel-Informationen sämtlicher Bildpunkte in der Reihenfolge im Bildschirmspeicher abgelegt sind, wie sie vom Elektronenstrahl auf den Bildschirm geschrieben werden. Beim Spectrum wird jedoch eine andere Organisationsform verwendet. Laut angabe von Hardware Designer Bruce Gordon wurde dies aus Gründen die etwas mit der ULA und den Z80 Registern zu tun hat, so angelegt wie wir es beim Spectrum kennen und - jetzt auch schon gewohnt sind.

Der erste Speicher-Hauptblock ist also in drei gleichorganisierte Teilbereiche aufgeteilt, die jeweils die Informationen von acht aufeinanderfolgenden Zeilen enthalten. Jeder Teilbereich ist so organisiert, daß zuerst das jeweils erste Byte jedes Zeichens der acht Zeilen des Teilbereichs gespeichert ist, dann das zweite Byte jedes Zeichens usw. bis zum achten Byte

Darauf folgen die ersten Bytes sämtlicher Zeichen in den acht Zeilen des nächsten Teilbereichs. Die Anwendung von PEEK und POKE wird dadurch natürlich komplizierter - ist aber nicht unmöglich!

Der zweite Hauptblock des D-Files enthält für jede PRINT-Position zusätzliche Informationen - die Attribute (FLASH, FARBE usw.)
Das zum Schluß dieses Beitrags abgedruckte Programm wirkt jedoch nur auf den ersten Block des Bildspeichers; Zusatzinformationen wie Bright Flash oder Farbe werden durch die Maschinencode-Routine nicht verändert.

Das M-Codeprogramm wurde für die 48K Spectrum Version geschrieben. Soll das Programm für die 16K-Version verwendet werden, sind für folgende Zahlen die Werte in den Klammern zu verwenden. 65267 (32499), 65268 (32500), 65360 (32592), 65269 (32501) und in allen DATA-Zeilen 254 (126), 255 (127); Ausnahme! Die erste 255 in DATA-Zeile 50 bleibt.

Die Nummer der zu löschenden Zeile (0 - 23) wird jetzt mit einem POKE-Befehl an die Adresse 65268 übergeben, und der Aufruf von RANDOMIZE USR 65269 startet das LöschProgramm. Zuerst erfolgt die Prüfung, ob die Zeilennummer den Wert 23 nicht überschreitet; ist dies der Fall, bricht das LöschProgramm ab und es wird im HauptProgramm mit dem folgenden Befehl weitergearbeitet. Fällt die Prüfung positiv aus werden die die Bildspeicheradressen der angegebenen Zeile errechnet und gelöscht. Danach wird in das Basic-Programm zurückgesprungen.

```
10 CLEAR 65267
20 FOR F=65268 TO 65360
30 READ A: POKE F,A: NEXT F
50 DATA 255,33,244,254,126,6,24,167,144,250,1,255,201,126,7,7,7,7
60 DATA 218,32,255,7,218,42,255,17,0,0,221,33,0,64,126,7,7,7,7,7,95
70 DATA 221,25,195,52,255,126,214,16,119,17,0,16,195,16,255,126,214,8
80 DATA 119,17,0,8,195,16,255,33,65,255,17,0,1,14,8,6,31,112,221,54
90 DATA0,0,5,250,74,255,195,62,255,13,200,221,25,195,60,255
```

Das folgende Beispiel zeigt ein Programm zum Löschen der Zeilen 8 bis 15 und anschließendem Einfügen einer Zeile mit dem Hash Character (#). Obwohl das MaschinencodeProgramm nach dem Löschen jeder Zeile in da BASIC-Programm zurückspringt, geht das Löschen etwa vierzig mal schneller als beim Verwenden eines reinen BASIC-Programms.

```
10 FOR F=1 TO 704
20 PRINT "0";
30 NEXT F
40 FOR N=8 TO 15
50 POKE 65268,N: RANDOMIZE USR 65269
60 NEXT N
70 PRINT AT 13,0;"############"
```

Das über RamtoP liegende Mcode-Programm kann mit dem Befehl SAVE "Ptcls" CODE 65268 (32500),93 gespeichert werden. Vor dem Laden des Programms muß durch CLEAR 65267 Platz über RamtoP geschaffen werden.

Noch etwas? - Nur viel Spaß und möglichst viele Programme in der diese nützliche Routine seinen Zweck erfüllt.

Klaus Ziereis
Ettmannsdorferstraße 119
8460 Schwandorf
09431/9369

Hallo Spektrumfreunde!

Ich habe folgende Hardware zu verkaufen:

1. Spektrum 48 K mit Iso-Rom 8.0	100,-
2. Zusatztastatur Saga 2001 mit Abdeckhaube	100,-
5. Sharp Datarecorder	40,-
6. Kempston E Centronics-Druckerinterface	60,-
8. ca. 10 Datasetten	10,-

Jede Menge Listings und Programmbeschreibungen und ASM-Pokes für 10,-

Außerdem verkaufe ich ASM-Hefte Nr.7/86 - 10/88 komplett für 40,-

Alle Preise auf Verhandlungsbasis!!

Hubert Kracher
Kirchdorfer Str. 5
8201 Großholzhausen

****
* DOUBLE 'H' SOFTWARE *
****

Harald Lack
Heidenauer Str. 5
8201 Raubling

Hallo allerseits!!

In unserem heutigen Beitrag zum Clubinfo wollen wir uns wieder an die Spielefreaks unter euch wenden. Wir haben uns für diesmal einen Vertreter aus dem Hause GREMLIN ausgesucht und zwar das Programm "JACK THE NIPPER II IN COCONUT CAPERS". Wie wohl jeder weiß, ist dies der nicht minder erfolgreiche Nachfolger von "JACK THE NIPPER" der bereits früher erschienen war. "Jack der II" stammt allerdings auch schon aus dem Jahre 1987 konnte allerdings damals wie heute durch seine Spielidee überzeugen. Obwohl gegenüber dem Vorgänger eigentlich von der Spielidee alles beim alten blieb, wurde das Game recht nett in Szene gesetzt. Jack befindet sich hier im Dschungel und hat mit allerlei Störenfrieden zu kämpfen. Wie GREMLIN im Inlay seines Spieles angibt, hat das Programm eigentlich kein Spielziel das erreicht werden muß. So gesehen kann man das Game auf zwei Arten spielen. Entweder man durchforstet den Dschungel und macht sich nur auf die Punktejagd oder man versucht mit einigen umherliegenden Gegenständen ein Paar der Einheimischen und wildlebende Tiere zu ärgern was natürlich wieder das aus dem ersten Teil übernommene Naughtyometer steigen läßt. Der Dschungel ist labyrinthartig aufgebaut und umfaßt annähernd einhundert Screens. Analog zum ersten Teil gibt es auch hier kein Scrolling sondern die Screens wechseln abrupt. Als sehr störend haben sich die Vögel erwiesen, die Jack oft das Leben sehr schwermachen. Allerdings ist Jack ihnen gegenüber nicht hilflos, denn er kann auch bis zu zwei Gegenstände aufnehmen um sie an verschiedenen Orten einzusetzen. Um in diesem Spiel auch nur annähernd eine Überlebenschance zu haben, sollte Jack nicht vergessen, eine Kokosnuß aufzunehmen die sich verstreut (wenn auch nicht in übermäßiger Zahl) im Dschungel finden läßt. Diese ermöglichen es ihm nämlich, nach seinen Feinden zu werfen und sie mit einem gezielten Treffer zu beseitigen. Leider ist die Zahl der möglichen Würfe pro Kokosnuß beschränkt, sodaß wildes um sich ballern nicht zu empfehlen ist. Wenn das Spiel gestartet wird, befindet sich Jack an einem der vorgegebenen Starträume und muß von dort aus, die Gegend erkunden. Besonders hübsch fanden wir übrigens die über mehrere Screens in die Höhe reichenden Bäume, die zum Klettern einladen. Allerdings muß man sehr aufpassen, um nicht unversehens abzustürzen und eines der kostbaren Leben auszuhauchen. Eine recht komplizierte Stelle im Programm bildet die lange Fahrt mit der Bergwerksbahn, da es hier nicht nur von Feinden wimmelt, sondern auch das richtige Timing gefunden werden muß. Es macht recht viel Spaß die Gegend zu erkunden, da die Screens recht anspruchsvoll gestaltet wurden. Hitunter ist die Orientierung in dem weitläufigen Dschungel etwas schwierig, sodaß es einem sicherlich hilft, wenn man seine Position schnell und einfach bestimmen kann. Beiliegend deshalb der von uns ausgearbeitete Plan der hoffentlich dem einen oder anderen weiterhilft.

Um sich nicht bereits nach wenigen Screens am Ende der zur Verfügung stehenden Leben zu befinden, hier noch eine kurze Laderoutine, die unendliche Leben bietet. Es braucht wohl nicht gesagt zu werden, daß die Wirksamkeit des Poke-Befehls von der jeweiligen Programmversion abhängt.

```
10 CLEAR 24575: LOAD"" SCREEN$
20 PRINT AT 17,0: LOAD""CODE
30 POKE 43251,0
40 RANDOMIZE USR 34240: REM UNENDLICHE LEBEN
```

Soviel für heute von unserer Seite. Bis demnächst an dieser Stelle.....

S1 - S3 = Startpositionen
X = Kokusnüsse
L = Zusatzleben
Jack the Nipper II
G = Sonstige Gegenstände
T = „Tarzan"
DOUBLE 'H.
Mapped by
SOFTWARE

Gregor Gellissen
Iltisstraße 15
4019 Monheim 1
Tel. 02173 / 54592

Um es vorwegzunehmen, möchte ich Ihnen sagen, daß ich gerne bereit bin auch im Hinblick auf weitere Hilfestellungen Ihrem Sinclair-User-Club beizutreten.
Es geht um folgendes:
Ich habe mir über die Zeitschrift Computerflohmarkt ein Kempston Centronics Interface für den Spectrum besorgt. Der Verkäufer hatte aber dezent verschwiegen, daß er nicht über die dazugehörende notwendige Software der Fa. Hilderbay Printer Software verfügt. Das Interface wird nämlich von der Fa. Kempston gemeinsam mit einer Softwarekassette dieser Fa. vertrieben.
Ich suche nun jemanden, der mir diese Softwarekassette auf eine Kassette kopiert. Es dürfte sich dabei um ein paar wenige Basic-Files und MC-Files handeln. Selbstverständlich würde ich dem Betreffenden nicht nur seine Unkosten, sondern auch ein Entgelt für die entstandene Mühe zukommen lassen. Anzumerken ist, daß ich die Programmversionen für den Spectrum 48K benötige. Vielleicht kennen Sie jemanden in Ihrem Club, der mir weiterhelfen kann. Hinsichtlich des Problems der Raubkopiererei mache ich mir keine Sorgen, da ich ja über die Hardware verfüge, zu der die Software gemeinsam (im Preis inbegriffen) geliefert wurde.
Das 2. Problem besteht darin, daß mir der gleiche Verkäufer einen ßDisk-Controller mit TR-DOS Version 4.09 angedreht hat, zu der die dazugehörende Utility-Diskette und das vollständige User-Manual fehlt. So bin ich zwar in der Lage bei dem Prompt A> und der Meldung * TR-DOS Ver 4.09 * c 1984 Technology Research Ltd. (U.K.) Befehle wie FORMAT, CAT etc. ausführen zu lassen, wobei meine 3,5-Zoll-Doppelfloppy gut funktioniert, jedoch schaffe ich es nicht(bei noch sichtbarem Prompt A>)ein Basic-Programm in den Spectrum einzugeben. Mit A> RETURN kehre ich dann zwar ins Basic zurück und ich kann das Programm eintippen, jedoch schaffe ich es nicht mehr, dieses Programm auf die Diskette zu speichern, da das A-Prompt anscheinend unwiederbringlich verloren ist. Jedenfalls habe ich bei allerlei Versuchen es nicht geschafft, das Programm zu speichern. Gibt es hierzu einen Trick oder fehlt ein Utility-File auf Diskette ? Auch funktioniert der sogenannte Magic-Button nicht, der dafür sorgt, daß die kompletten 48K-Ram meines Spectrum auf Diskette abgespeichert werden. Drücke ich nämlich nach Re-Setten des ßDisk-Interfaces den Befehl CAT (wobei die Diskette zuvor formatiert wurde und über 2544 freie Sektoren verfügt), erscheint die Meldung A: (C)192, was bedeutet, daß 192 Sektoren gleich 48K belegt wurden (in Maschinencode, daher das C). Dieses Magic-Button-File sollte aber den Namen "@" (d.h. Klammeraffe) haben, was offensichtlich nicht der Fall ist. Hingegen scheinen die 48K mit " " (d.h. 1 Blank) abgespeichert worden zu sein. Gebe ich nämlich den Befehl A> GOTO " " CODE ein, wird dieses File geladen. An sich sollte es damit den Speicher mit 48K so wiederherstellen, wie es beim Abspeichern mit dem Magic-Button war und dementsprechend weiterlaufen, aber es erscheint nur ein bunt flimmerndes Bild auf dem Monitor und der Spectrum stürzt ab, bzw. re-settet sich selbst.
Auch bitte ich Sie um Hilfe, vielleicht gibt es jemanden in Ihrem Club, der auch das ßDisk-Interface benutzt und weiß, wie es richtig angesprochen werden muß (ggf. unter Zuhilfenahme der o.g. Utility-Diskette, welche mir fehlt). Selbstverständlich können Sie meine volle Anschrift und Telefonnummer weitergeben. Auch hier würde ich demjenigen, der mir weiterhelfen kann, seine Unkosten und Mühewaltung großzügig vergüten.

Ich hoffe mit diesem umfangreichen Brief Ihre Zeit nicht zu sehr in Anspruch genommen zu haben und verbleibe mit den besten Grüßen

Spectrum User Club Wuppertal

Verkaufe:

1 SPECTRUM +

1 Netzteil f. SPECTRUM

1 Datenrecorder MC 3810

Tasword 2 incl.Taspoke u. Multitas

Div. andere Software

1 Drucker Shinwa CP 80 (defekt)

1 Druckerinterface Kempston E

alles zus. DM 200

U.Dünow
Matth.-Claud. Str.23
4407 Emsdetten
Tel. 02572/83320 (ab 16 Uhr)
02572/3095 (bis 6 Uhr)

Gerold Windt
Waldstr. 15

6054 Rodgau 6

(06106) 44 76

Rodgau 6, 15.01.90

Hallo Spectrum User !

Folgende Sachen biete ich an:

Bücher je 10 DM:

Astronomie ZX Spectrum, Computer Lösungen Für Schule und Studium, ZX Spectrum Hardware Manuall in Deutsch, Mikrodrive Handbuch (Jochen Merz), Das Mikrodrive Universum (Logan), Spass und Profit, 60 Programme für den Spectrum (Huber), ZX ohne Grenzen, Das Sinclair Spectrum Rom, Das Spectrum Buch, ZX Spectrum (Hartnell), Rund um den Spectrum (Flögel).

30 DM Joystickinterface mit 2 Ports

Biete jede Menge orig Anleitungen (Hisoft c, Pascal, Compiler usw.). Viele Kleinteile:(Busverlängerungen, Stecker, Kabel usw.) Ein Anruf lohnt sich immer ! 06106/4476

100 DM Multiface 128

220 DM Die letzten SUPER EPROMMER Menügesteuert Eproms bis 27256. Epromm er läßt sich als Epromm floppy verwenden. Eigenes Betriebssystem!!!(Textoolsockel)

VB DM mehrere Joystickinterfaces (progr. und einstell.)

VB mehrere Spectrum und Discovery Platinen für Bastler

Tel:06106/4476
Repariere Sinclair Geräte aller Art (auch Discovery), auf wunsch mit Kostenvoranschlag. Suche ständig Sinclairschrott gegen Bezahlung.